

driver and the bus or network interface driver communicate through a vendor-specific API because both drivers are written by the manufacturer of the network device being accessed. Therefore, while the NDIS miniport must conform to the NDIS API in order to communicate with NDIS layer 116, and the bus or network interface must conform to the appropriate bus or network driver in passing information to network device, the interaction between the NDIS miniport and the bus or network interface is completely at the discretion of the hardware manufacturer.

A/ Requiring device manufacturers to write two drivers for each piece of equipment they market presents some fairly serious problems. For example, the sheer number of device drivers is difficult and expensive to manage, both for the hardware manufacturer and for operating system developers who may distribute certain device drivers with their software. Furthermore, since manufacturers provide both the connection to NDIS and the bus or network interface, the network functionality and the specifics of a particular bus are likely to be coupled, making it impossible to update one without the other. Solving these problems will allow for faster deployment of remotely connected networking devices and lower costs for developing host-based drivers.

The NDIS miniport driver and the bus or network interface, both of which were provided by the device hardware manufacturer, can be replaced with a Remote NDIS miniport layer and bus- or network-specific microports. The Remote NDIS miniport layer and bus- or network-specific microports are independent of the particular device being accessed and therefore can be included as part of the operating system. Therefore, hardware manufacturers writing to the Remote NDIS specification are no longer required to write host-based drivers for their devices. Remote NDIS defines a connection-agnostic or connection-independent message set along with a description of how the message set operates over a particular connection, such as a specific bus or network. Because the remote NDIS interface is standardized, a core set of host drivers can

support any number of attached networking devices, thereby improving system stability and user satisfaction because no new drivers must be installed to support a new network device. The remote NDIS architecture includes a remote NDIS miniport driver that understands the remote NDIS message set and communicates with bus- or network-specific microport drivers. Specifically, the Remote NDIS miniport layer encapsulates NDIS OIDs (Object Identifiers) and NDIS data packets into data structures that can be passed without modification to a networking device. The data structures are known as Remote NDIS messages.

A/ The bus- or network-specific microport drivers represent an intermediate layer that understands the bus or network responsible for passing the messages onto the device. Thus, the microport layer receives the remote NDIS messages and passes them to the corresponding element of a bus or network driver layer. The bus or network driver layer then passes the remote NDIS message to remote the NDIS devices.

Because network protocol mechanisms are abstracted above the bus or network-specific microport layer, adding new network functionality can be accomplished by changing only the remote NDIS miniport layer. The microport layer remains unchanged because it is merely a message transport mechanism that passes NDIS OIDs and NDIS data packets encapsulated in remote NDIS messages. Furthermore, adding network functionality in the form of new NDIS OIDs is available to all bus or network microports because a single remote NDIS miniport layer can serve them all. The present invention also maintains backward compatibility. As new NDIS OIDs are added, a remote NDIS device may respond that it does not understand the NDIS OID and therefore does not support the new network functionality."

Please replace the paragraph beginning on page 12, line 1 with the following paragraph:

A2
Network data is passed between the host 164 and the device 162 over the L2CAP channel 150. This data can be encapsulated in an NDIS packet mechanism, on the model already used by the NDIS network stack. The maximum length of packets supported by L2CAP can be the maximum MTU of the media minus the RNIDS header size. The device 152 can fill in the MaxTransferSize value in an NDIS function call to the largest L2CAP message it can send. If the host 164 has a smaller L2CAP maximum message size, it can overwrite the returned information with its own maximum message size. Either the host 164 or the device 162 can initiate the setup of both the control and data L2CAP channels.

Please replace the paragraph beginning on ~~page 12~~, line 10 with the following paragraph:

A3
A minimal Service Discovery Protocol (SDP) record which can be used for a Bluetooth Remote NDIS device can be as shown in Table 1 below. As can be seen, the Remote NDIS device uses the standard Service Discovery description. Personal Area Network (PAN) services can communicate with each other. It is possible for a Bluetooth device to have multiple PAN services. For example, a cellular phone can have a Wireless WAN server that gives Bluetooth devices access to the cellular data network. In such a case the ServiceName can be "WWAN", or an even more descriptive name. Alternatively, the cellular phone can have a PAN service that allows internal PAN service to communicate peer-to-peer between devices. In such a case, the ServiceName can be set to "PEER". A device should not advertise more than one PAN profile with a ServiceName of PEER.

Please replace the paragraph beginning on ~~page 14~~, line 14 with the following paragraph:

A4
Remote NDIS defines the format for the REMOTE_NDIS_PACKET message including space to transport NDIS OOB and per packet information fields. The per packet information

AL
files can be supplied by NDIS when the remote driver specifies it supports the functionality. The OOB information can be supported for particular media types. For example, for Ethernet peer-to-peer emulation fields are not required. In such a case, 44 bytes of offsets and lengths of packet information fields are not required. Thus, for a 1514 byte Ethernet MTU L2CAP can have a minimum of 1554 byte MTU and wastes approximately 3% of the Bluetooth bandwidth. This could be an issue for slow links. This can be optimized if DataOffset is 4, assuming the rest of the RNDIS_PACKET header is NULL. This can reduce the data overhead to 16 bytes or 1%.

Please replace the paragraph beginning on page 15, line 1 with the following paragraph:

AS
In operation, the Bluetooth microport can be loaded when a remote Bluetooth device, which supports the PAN service, is in range and can be communicated to. Loading the microport can cause an initialization message to be sent from the microport. When the remote device is a cellular phone, for example, it can respond with an initialization complete message. When the remote device is another Windows machine, as another example, it can also generate an initialization message. The microport can also receive extra messages such as REMOTE_NDIS_INITIALIZE_MSG, REMOTE_NDIS_QUERY_MSG, REMOTE_NDIS_SET_MSG, REMOTE_NDIS_RESET_MSG, and REMOTE_NDIS_KEEPALIVE_MSG. These messages are described in more detail in the article entitled "Remote NDIS Over Bluetooth Specification", dated March 20, 2000, and attached at Appendix F.
